COMPSCI 514: Algorithms for Data Science

Cameron Musco

University of Massachusetts Amherst. Spring 2026.

Lecture 3

Logistics

- Problem Set 1 was posted on Friday (see Assignments Tab on course webpage) and is due **Friday 2/20 at 11:59pm**.
- The 'Challenge Problems' are optional.
- On the quiz many number of people cited concerns about their probability/stats background. We will be going over background material for the next few classes, so try to stay on top of things these first few weeks, and you should be ok.
- For linear algebra you have time to review – we will only use it after the first midterm.
- Stop by office hours for probability review, to go over material from lecture, etc. I also highly recommend the exercises in *Foundations of Data Science* and *Probability and Computing*.
- It is common to not catch everything in lecture. I strongly encourage going back to the slides to review, and to interrupt with questions during class.

Content Overview

Last Class:

- ~~Linearity of variance.~~
- Markov's inequality: the most fundamental concentration bound. $\Pr(X \geq t \cdot \mathbb{E}[X]) \leq 1/t$.
- Algorithmic applications of Markov's inequality, linearity of expectation, and indicator random variables:
  - Counting collisions to estimate CAPTCHA database size.
  - Analysis of hash tables with random hash functions.
  - Collisions free hashing using a table with $O(m^2)$ slots to store m items.

Content Overview

Today:

- 2-level hashing for optimal lookup time and storage.
- 2-universal and pairwise independent hash functions.
- Start on application of random hashing to distributed load balancing.
- Through this application learn about Chebyshev's inequality, which strengthens Markov's inequality.

5 1 point

The expected number of inches of rain on Saturday is 5.8 and the expected number of inches on Sunday is 6.9. ~~There is a 50% chance of rain on Saturday. If it rains on Saturday, there is a 75% chance of rain on Sunday. If it does not rain on Saturday, there is only a 25% chance of rain on Sunday.~~ What is the expected number of inches of rainfall total over the weekend?

Type your answer...

$$\mathbb{E}[\text{Total}] = \mathbb{E}[\text{Sat} + \text{Sun}]$$

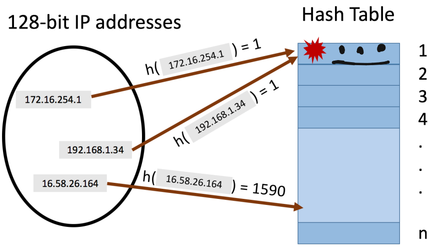
$$= \mathbb{E}[\text{Sat}] + \mathbb{E}[\text{Sun}]$$

$$= 5.8 + 6.9 = 12.7$$

Quiz Questions

Hash Tables

We store m items from a large universe in a hash table with n positions.



- Want to show that when $\mathsf{h} : U \to [n]$ is a fully random hash function, query time is $O(1)$ with good probability.
- Equivalently: want to show that there are few collisions between hashed items.

Collision Free Hashing

Let $\mathbf{C} = \sum_{i,j \in [m], i<j} \mathbf{C}_{i,j}$ be the number of pairwise collisions between items.

Cindicator if item i and j hash to same bucket.

$$\mathbb{E}[\mathbf{C}] = \frac{m(m-1)}{2n} \quad \text{(via the Captcha analysis)}$$

$$\binom{m}{2} \cdot \frac{1}{n}$$

m: total number of stored items, n: hash table size, $\mathbf{C}$: total pairwise collisions in table.

Collision Free Hashing

Let $\mathsf{C} = \sum_{i,j \in [m], i<j} \mathsf{C}_{i,j}$ be the number of pairwise collisions between items.

$$\mathbb{E}[\mathsf{C}] = \frac{m(m-1)}{2n} \quad \text{(via the Captcha analysis)}$$

- For $n = 4m^2$ we have: $\mathbb{E}[\mathsf{C}] = \frac{m(m-1)}{8m^2} \leq \frac{1}{8}$.

m: total number of stored items, *n*: hash table size, C: total pairwise collisions in table.

Collision Free Hashing

Let $\mathsf{C} = \sum_{i,j \in [m], i<j} \mathsf{C}_{i,j}$ be the number of pairwise collisions between items.

$$\mathbb{E}[\mathsf{C}] = \frac{m(m-1)}{2n} \quad \text{(via the Captcha analysis)}$$

- For $n = 4m^2$ we have: $\mathbb{E}[\mathsf{C}] = \frac{m(m-1)}{8m^2} \leq \frac{1}{8}$.

Apply Markov's Inequality: $\Pr[\mathsf{C} \geq 1] \leq \frac{\mathbb{E}[\mathsf{C}]}{1} = \frac{1}{8}$.

m: total number of stored items, n: hash table size, C: total pairwise collisions in table.

Collision Free Hashing

Let $\mathsf{C} = \sum_{i,j \in [m], i<j} \mathsf{C}_{i,j}$ be the number of pairwise collisions between items.

$$\mathbb{E}[\mathsf{C}] = \frac{m(m-1)}{2n} \quad \text{(via the Captcha analysis)}$$

- For $n = 4m^2$ we have: $\mathbb{E}[\mathsf{C}] = \frac{m(m-1)}{8m^2} \leq \frac{1}{8}$.

Apply Markov's Inequality: $\Pr[\mathsf{C} \geq 1] \leq \frac{\mathbb{E}[\mathsf{C}]}{1} = \frac{1}{8}$.

$$\Pr[\mathsf{C} = 0] = 1 - \Pr[\mathsf{C} \geq 1]$$

m: total number of stored items, n: hash table size, C: total pairwise collisions in table.

Collision Free Hashing

Let $\mathbf{C} = \sum_{i,j \in [m], i<j} \mathbf{C}_{i,j}$ be the number of pairwise collisions between items.

$$\mathbb{E}[\mathbf{C}] = \frac{m(m-1)}{2n} \quad \text{(via the Captcha analysis)}$$

- For $n = 4m^2$ we have: $\mathbb{E}[\mathbf{C}] = \frac{m(m-1)}{8m^2} \leq \frac{1}{8}$.

Apply Markov's Inequality: $\Pr[\mathbf{C} \geq 1] \leq \frac{\mathbb{E}[\mathbf{C}]}{1} = \frac{1}{8}$.

$$\Pr[\mathbf{C} = 0] = 1 - \Pr[\mathbf{C} \geq 1] \geq 1 - \frac{1}{8} = \frac{7}{8}$$

m: total number of stored items, n: hash table size, $\mathbf{C}$: total pairwise collisions in table.

Let $\mathbf{C} = \sum_{i,j \in [m], i<j} \mathbf{C}_{i,j}$ be the number of pairwise collisions between items.

$$\mathbb{E}[\mathbf{C}] = \frac{m(m-1)}{2n} \quad \text{(via the Captcha analysis)}$$

$n = m^2$ $\quad$ $\mathbb{E}[C] = \frac{m(m-1)}{2m^2} \leq \frac{1}{2}$

- For $n = 4m^2$ we have: $\mathbb{E}[\mathbf{C}] = \frac{m(m-1)}{8m^2} \leq \frac{1}{8}$.

nothing special about 4.

Apply Markov's Inequality: $\Pr[\mathbf{C} \geq 1] \leq \frac{\mathbb{E}[\mathbf{C}]}{1} = \frac{1}{8}$.

$$\Pr[\mathbf{C} = 0] = 1 - \Pr[\mathbf{C} \geq 1] \geq 1 - \frac{1}{8} = \frac{7}{8}$$

I.e., with probability at least 7/8 we have no collisions and thus $O(1)$ query time. But we are using $O(m^2)$ space to store m items...

> m: total number of stored items, n: hash table size, $\mathbf{C}$: total pairwise collisions in table.

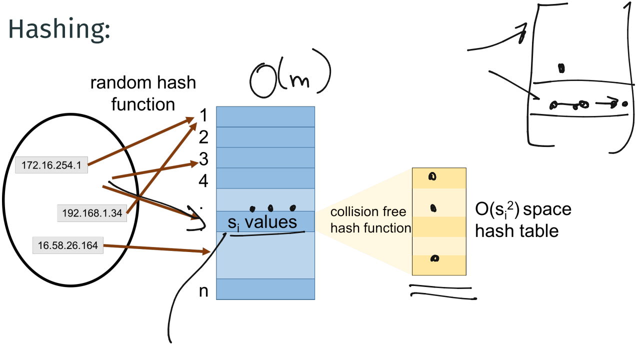
Two Level Hashing

Want to preserve $O(1)$ query time while using $O(m)$ space.

Two Level Hashing

Want to preserve $O(1)$ query time while using $O(m)$ space.

Two-Level Hashing:

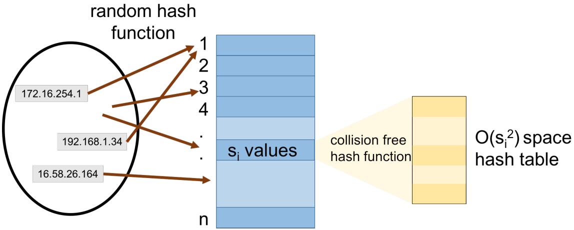


S_i is a random variable

but we know its value

Want to preserve $O(1)$ query time while using $O(m)$ space.

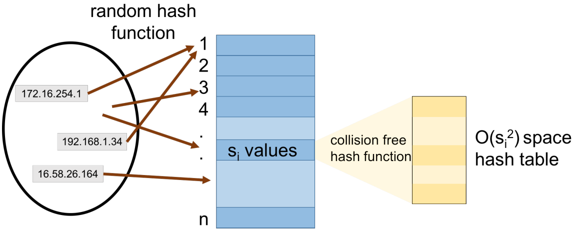
Two-Level Hashing:



- For each bucket with s_i values, pick a collision free hash function mapping $[s_i] \to [s_i^2]$.

Want to preserve $O(1)$ query time while using $O(m)$ space.

Two-Level Hashing:



- For each bucket with s_i values, pick a collision free hash function mapping $[s_i] \to [s_i^2]$.
- **Just Showed:** A random function is collision free with probability $\geq \frac{7}{8}$ so can just generate a random hash function and check if it is collision free.

Space Usage

Query time for two level hashing is $O(1)$: requires evaluating two hash functions.

x_j, x_k: stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored in hash table at position i.

Space Usage

Query time for two level hashing is $O(1)$: requires evaluating two hash functions. What is the expected space usage?

$O(m)$ to store m items.

x_j, x_k: stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored in hash table at position i.

Space Usage

Query time for two level hashing is $O(1)$: requires evaluating two hash functions. What is the expected space usage?

Up to constants, space used is: $\mathbf{S} = n + \sum_{i=1}^{n} \mathbf{s}_i^2$

$\mathbb{E}S = O(m)$

first table

total size of backup tables

x_j, x_k: stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored in hash table at position i.

Space Usage

Query time for two level hashing is $O(1)$: requires evaluating two hash functions. What is the expected space usage?

Up to constants, space used is: $\mathbb{E}[\mathbf{S}] = n + \sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2]$

x_j, x_k: stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored in hash table at position i.

Space Usage

Query time for two level hashing is $O(1)$: requires evaluating two hash functions. What is the expected space usage?

$\boxed{n = 2m}$

Up to constants, space used is: $\mathbb{E}[\mathbf{S}] = n + \sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2]$

$(\#\text{ items in bucket } i)^2$

$s_1 = 0$

$s_2 = 1$

$s_{10} = 3$

s_{100}

$0 \leq s_i \leq m$

x_j, x_k: stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored in hash table at position i.

Space Usage

Query time for two level hashing is $O(1)$: requires evaluating two hash functions. What is the expected space usage?

Up to constants, space used is: $\mathbb{E}[\mathbf{S}] = n + \sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2]$

$$\boxed{\mathbb{E}[\mathbf{s}_i^2]} = \mathbb{E}\left[\left(\underbrace{\sum_{j=1}^{m} \mathbb{I}_{\mathbf{h}(x_j)=i}}_{s_i}\right)^2\right]$$

item j is in bucket i

Space used by i^{th} backup table.

$$\mathbb{E}[s_i] = \frac{m}{n}$$

$$\cancel{\mathbb{E}[s_i^2] = \left(\frac{m}{n}\right)^2}$$

x_j, x_k: stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored in hash table at position i.

Space Usage

Query time for two level hashing is $O(1)$: requires evaluating two hash functions. What is the expected space usage?

Up to constants, space used is: $\mathbb{E}[\mathbf{S}] = n + \sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2]$

$$\mathbb{E}[\mathbf{s}_i^2] = \mathbb{E}\left[\left(\sum_{j=1}^{m} \mathbb{I}_{\mathbf{h}(x_j)=i}\right)^2\right]$$

$$= \mathbb{E}\left[\sum_{j,k\in[m]} \underbrace{\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}}\right]$$

Collisions again!

$= 1$ iff $h(x_j) = h(x_k) = i$

i.e. if items j and k collide in bucket i

x_j, x_k: stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored in hash table at position i.

Space Usage

Query time for two level hashing is $O(1)$: requires evaluating two hash functions. What is the expected space usage?

Up to constants, space used is: $\mathbb{E}[\mathbf{S}] = n + \sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2]$

$$\mathbb{E}[\mathbf{s}_i^2] = \mathbb{E}\left[\left(\sum_{j=1}^{m} \mathbb{I}_{\mathbf{h}(x_j)=i}\right)^2\right]$$

$$= \mathbb{E}\left[\sum_{j,k\in[m]} \mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \sum_{j,k\in[m]} \mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right].$$

x_j, x_k: stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored in hash table at position i.

Space Usage

Query time for two level hashing is $O(1)$: requires evaluating two hash functions. What is the expected space usage?

Up to constants, space used is: $\mathbb{E}[\mathbf{S}] = n + \sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2]$

$$\mathbb{E}[\mathbf{s}_i^2] = \mathbb{E}\left[\left(\sum_{j=1}^{m} \mathbb{I}_{\mathbf{h}(x_j)=i}\right)^2\right]$$

$$= \mathbb{E}\left[\sum_{j,k\in[m]} \mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \sum_{j,k\in[m]} \mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right].$$

- For $j = k$, $\mathbb{E}\Big[\underbrace{\mathbb{I}_{h(x_j)=i} \cdot \mathbb{I}_{h(x_j)=i}}_{1 \text{ if } h(x_j)=i \text{ or } 0 \text{ otherwise}}\Big] = \mathbb{E}\left[\mathbb{I}^2_{h(x_j)=i}\right] = \mathbb{E}\left[\mathbb{I}_{h(x_j)=i}\right] = \frac{1}{n}$

x_j, x_k: stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored in hash table at position i.

Space Usage

Query time for two level hashing is $O(1)$: requires evaluating two hash functions. What is the expected space usage?

Up to constants, space used is: $\mathbb{E}[\mathbf{S}] = n + \sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2]$

$$\mathbb{E}[\mathbf{s}_i^2] = \mathbb{E}\left[\left(\sum_{j=1}^{m} \mathbb{I}_{\mathbf{h}(x_j)=i}\right)^2\right]$$

$$= \mathbb{E}\left[\sum_{j,k\in[m]} \mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \sum_{j,k\in[m]} \mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right].$$

- For $j = k$,

$$\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \mathbb{E}\left[\left(\mathbb{I}_{\mathbf{h}(x_j)=i}\right)^2\right]$$

x_j, x_k: stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored in hash table at position i.

Space Usage

Query time for two level hashing is $O(1)$: requires evaluating two hash functions. What is the expected space usage?

Up to constants, space used is: $\mathbb{E}[\mathbf{S}] = n + \sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2]$

$$\mathbb{E}[\mathbf{s}_i^2] = \mathbb{E}\left[\left(\sum_{j=1}^{m} \mathbb{I}_{\mathbf{h}(x_j)=i}\right)^2\right]$$

$$= \mathbb{E}\left[\sum_{j,k\in[m]} \mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \sum_{j,k\in[m]} \mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right].$$

- For $j = k$,

$$\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \mathbb{E}\left[\left(\mathbb{I}_{\mathbf{h}(x_j)=i}\right)^2\right] = \Pr[\mathbf{h}(x_j) = i]$$

x_j, x_k: stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored in hash table at position i.

Space Usage

Query time for two level hashing is $O(1)$: requires evaluating two hash functions. What is the expected space usage?

Up to constants, space used is: $\mathbb{E}[\mathbf{S}] = n + \sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2]$

$$\mathbb{E}[\mathbf{s}_i^2] = \mathbb{E}\left[\left(\sum_{j=1}^{m} \mathbb{I}_{\mathbf{h}(x_j)=i}\right)^2\right]$$

$$= \mathbb{E}\left[\sum_{j,k\in[m]} \mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \sum_{j,k\in[m]} \mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right].$$

- For $j = k$,

$$\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \mathbb{E}\left[\left(\mathbb{I}_{\mathbf{h}(x_j)=i}\right)^2\right] = \Pr[\mathbf{h}(x_j) = i] = \frac{1}{n}.$$

x_j, x_k: stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored in hash table at position i.

Space Usage

Query time for two level hashing is $O(1)$: requires evaluating two hash functions. What is the expected space usage?

Up to constants, space used is: $\mathbb{E}[\mathbf{S}] = n + \sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2]$

$$\mathbb{E}[\mathbf{s}_i^2] = \mathbb{E}\left[\left(\sum_{j=1}^{m} \mathbb{I}_{\mathbf{h}(x_j)=i}\right)^2\right]$$

$$= \mathbb{E}\left[\sum_{j,k\in[m]} \mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \sum_{j,k\in[m]} \mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right].$$

- For $j = k$,

  $\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \mathbb{E}\left[\left(\mathbb{I}_{\mathbf{h}(x_j)=i}\right)^2\right] = \Pr[\mathbf{h}(x_j) = i] = \frac{1}{n}.$

  1 or 1 if x_j and x_k both hash to i

- For $j \neq k$,

x_j, x_k: stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored in hash table at position i.

Space Usage

Query time for two level hashing is $O(1)$: requires evaluating two hash functions. What is the expected space usage?

Up to constants, space used is: $\mathbb{E}[\mathbf{S}] = n + \sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2]$

$$\mathbb{E}[\mathbf{s}_i^2] = \mathbb{E}\left[\left(\sum_{j=1}^{m} \mathbb{I}_{\mathbf{h}(x_j)=i}\right)^2\right]$$

$$= \mathbb{E}\left[\sum_{j,k\in[m]} \mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \sum_{j,k\in[m]} \mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right].$$

- For $j = k$,
  $\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \mathbb{E}\left[\left(\mathbb{I}_{\mathbf{h}(x_j)=i}\right)^2\right] = \Pr[\mathbf{h}(x_j) = i] = \frac{1}{n}.$
- For $j \neq k$, $\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right]$

x_j, x_k: stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored in hash table at position i.

Space Usage

Query time for two level hashing is $O(1)$: requires evaluating two hash functions. What is the expected space usage?

Up to constants, space used is: $\mathbb{E}[\mathbf{S}] = n + \sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2]$

$$\mathbb{E}[\mathbf{s}_i^2] = \mathbb{E}\left[\left(\sum_{j=1}^{m} \mathbb{I}_{\mathbf{h}(x_j)=i}\right)^2\right]$$

$\mathbb{E}[\mathbb{I}_j \cdot \mathbb{I}_k]$

$\mathbb{I}_j$: 1 if $h(x_j)=i$, 0 otherwise

$\mathbb{I}_k$: 1 if $h(x_k)=i$, 0 otherwise

$$= \mathbb{E}\left[\sum_{j,k\in[m]} \mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \sum_{j,k\in[m]} \mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right].$$

- For $j = k$,
$$\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \mathbb{E}\left[\left(\mathbb{I}_{\mathbf{h}(x_j)=i}\right)^2\right] = \Pr[\mathbf{h}(x_j) = i] = \frac{1}{n}.$$
- For $j \neq k$, $\mathbb{E}\Big[\underbrace{\mathbb{I}_{\mathbf{h}(x_j)=i}}_{\mathbb{I}_j} \cdot \underbrace{\mathbb{I}_{\mathbf{h}(x_k)=i}}_{\mathbb{I}_k}\Big] = \Pr[\mathbf{h}(x_j) = i \cap \mathbf{h}(x_k) = i] = \frac{1}{n^2}$

x_j, x_k: stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored in hash table at position i.

Space Usage

Query time for two level hashing is $O(1)$: requires evaluating two hash functions. What is the expected space usage?

Up to constants, space used is: $\mathbb{E}[\mathbf{S}] = n + \sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2]$

$$\mathbb{E}[\mathbf{s}_i^2] = \mathbb{E}\left[\left(\sum_{j=1}^{m} \mathbb{I}_{\mathbf{h}(x_j)=i}\right)^2\right]$$

$$= \mathbb{E}\left[\sum_{j,k\in[m]} \mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \sum_{j,k\in[m]} \mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right].$$

- For $j = k$,
  $\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \mathbb{E}\left[\left(\mathbb{I}_{\mathbf{h}(x_j)=i}\right)^2\right] = \Pr[\mathbf{h}(x_j) = i] = \frac{1}{n}$.
- For $j \neq k$, $\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \Pr[\mathbf{h}(x_j) = i \cap \mathbf{h}(x_k) = i] = \frac{1}{n^2}$.

x_j, x_k: stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored in hash table at position i.

Space Usage

$$\mathbb{E}[\mathsf{s}_i^2] = \sum_{j,k \in [m]} \mathbb{E}\left[\mathbb{I}_{\mathsf{h}(x_j)=i} \cdot \mathbb{I}_{\mathsf{h}(x_k)=i}\right]$$

Space usage
of backup hash
table.

- For $j = k$, $\mathbb{E}\left[\mathbb{I}_{\mathsf{h}(x_j)=i} \cdot \mathbb{I}_{\mathsf{h}(x_k)=i}\right] = \frac{1}{n}$.
- For $j \neq k$, $\mathbb{E}\left[\mathbb{I}_{\mathsf{h}(x_j)=i} \cdot \mathbb{I}_{\mathsf{h}(x_k)=i}\right] = \frac{1}{n^2}$.

x_j, x_k: stored items, m: # stored items, n: hash table size, h: random hash function, S: space usage of two level hashing, s_i: # items stored at pos i.

Space Usage

$$\mathbb{E}[\mathbf{s}_i^2] = \sum_{j,k \in [m]} \mathbb{E}\left[\overset{1/n}{\mathbb{I}_{\mathbf{h}(x_j)=i}} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right]$$

$$= m \cdot \frac{1}{n} + 2 \cdot \underbrace{\binom{m}{2}}_{m(m-1)} \cdot \frac{1}{n^2}$$

m^2 terms total

m of them have $j=k$

$m(m-1)$ have $j \neq k$

- For $j = k$, $\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \frac{1}{n}$.
- For $j \neq k$, $\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \frac{1}{n^2}$.

x_j, x_k: stored items, m: # stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored at pos i.

Space Usage

$$\mathbb{E}[\mathbf{s}_i^2] = \sum_{j,k \in [m]} \mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right]$$
$$= m \cdot \frac{1}{n} + 2 \cdot \binom{m}{2} \cdot \frac{1}{n^2}$$

- For $j = k$, $\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \frac{1}{n}$.
- For $j \neq k$, $\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \frac{1}{n^2}$.

x_j, x_k: stored items, m: # stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored at pos i.

Space Usage

$$\mathbb{E}[\mathbf{s}_i^2] = \sum_{j,k \in [m]} \mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right]$$
$$= m \cdot \frac{1}{n} + 2 \cdot \binom{m}{2} \cdot \frac{1}{n^2}$$

- For $j = k$, $\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \frac{1}{n}$.
- For $j \neq k$, $\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \frac{1}{n^2}$.

x_j, x_k: stored items, m: # stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored at pos i.

Space Usage

$$\mathbb{E}[\mathsf{s}_i^2] = \sum_{j,k \in [m]} \mathbb{E}\left[\mathbb{I}_{\mathsf{h}(x_j)=i} \cdot \mathbb{I}_{\mathsf{h}(x_k)=i}\right]$$
$$= m \cdot \frac{1}{n} + 2 \cdot \binom{m}{2} \cdot \frac{1}{n^2}$$
$$= \frac{m}{n} + \frac{m(m-1)}{n^2}$$

- For $j = k$, $\mathbb{E}\left[\mathbb{I}_{\mathsf{h}(x_j)=i} \cdot \mathbb{I}_{\mathsf{h}(x_k)=i}\right] = \frac{1}{n}$.
- For $j \neq k$, $\mathbb{E}\left[\mathbb{I}_{\mathsf{h}(x_j)=i} \cdot \mathbb{I}_{\mathsf{h}(x_k)=i}\right] = \frac{1}{n^2}$.

x_j, x_k: stored items, m: # stored items, n: hash table size, h: random hash function, S: space usage of two level hashing, s_i: # items stored at pos i.

Space Usage

$$\mathbb{E}[\mathbf{s}_i^2] = \sum_{j,k\in[m]} \mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right]$$

$$= m \cdot \frac{1}{n} + 2 \cdot \binom{m}{2} \cdot \frac{1}{n^2}$$

$$= \frac{m}{n} + \frac{m(m-1)}{n^2} \leq 2 \text{ if we set } \boxed{n = m.}$$

- For $j = k$, $\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \frac{1}{n}$.
- For $j \neq k$, $\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \frac{1}{n^2}$.

x_j, x_k: stored items, m: # stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored at pos i.

Space Usage

$$\mathbb{E}[\mathbf{s}_i^2] = \sum_{j,k \in [m]} \mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right]$$
$$= m \cdot \frac{1}{n} + 2 \cdot \binom{m}{2} \cdot \frac{1}{n^2}$$
$$= \frac{m}{n} + \frac{m(m-1)}{n^2} \leq 2 \text{ if we set } n = m.$$

- For $j = k$, $\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \frac{1}{n}$.
- For $j \neq k$, $\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \frac{1}{n^2}$.

Total Expected Space Usage: (if we set $\boxed{n = m}$)

$$\mathbb{E}[\mathbf{S}] = n + \sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2]$$

x_j, x_k: stored items, m: # stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored at pos i.

Space Usage

$$\mathbb{E}[\mathbf{s}_i^2] = \sum_{j,k \in [m]} \mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right]$$

$$= m \cdot \frac{1}{n} + 2 \cdot \binom{m}{2} \cdot \frac{1}{n^2}$$

$$= \frac{m}{n} + \frac{m(m-1)}{n^2} \leq 2 \text{ if we set } n = m.$$

- For $j = k$, $\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \frac{1}{n}$.
- For $j \neq k$, $\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \frac{1}{n^2}$.

Total Expected Space Usage: (if we set $n = m$)

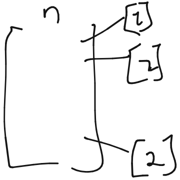
#items

$$\mathbb{E}[\mathbf{S}] = n + \sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2] \leq n + n \cdot 2 = 3n = 3m.$$

$O(m)$

x_j, x_k: stored items, m: # stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored at pos i.

Space Usage



$$\mathbb{E}[\mathbf{s}_i^2] = \sum_{j,k\in[m]} \mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right]$$

$$= m \cdot \frac{1}{n} + 2 \cdot \binom{m}{2} \cdot \frac{1}{n^2}$$

$$= \frac{m}{n} + \frac{m(m-1)}{n^2} \leq 2 \text{ if we set } n = m.$$

$m = n$

$(1,1)$
$(1,2)$
$(1,3)$
$(3,1)$

- For $j = k$, $\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \frac{1}{n}$.
- For $j \neq k$, $\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \frac{1}{n^2}$.

$\binom{m}{2} : \frac{m(m-1)}{2}$

Total Expected Space Usage: (if we set $n = m$)

$$\mathbb{E}[\mathbf{S}] = n + \sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2] \leq n + n \cdot 2 = 3n = 3m.$$

Near optimal space with $O(1)$ query time!

$(a+b+c)(a+b+c)$
$a^2+b^2+c^2+2ab+2bc+2ca$

x_j, x_k: stored items, m: # stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored at pos i.

Efficiently Computable Hash Function

So Far: we have assumed a **fully random hash function** $\mathbf{h}(x)$ with $\Pr[\mathbf{h}(x) = i] = \frac{1}{n}$ for $i \in 1, \ldots, n$ and $\mathbf{h}(x), \mathbf{h}(y)$ independent for $x \neq y$.

$$\Pr[h(x_j) = i \cap h(x_k) = i] = \frac{1}{n^2}$$

Efficiently Computable Hash Function

So Far: we have assumed a **fully random hash function** $\mathrm{h}(x)$ with $\Pr[\mathrm{h}(x) = i] = \frac{1}{n}$ for $i \in 1, \ldots, n$ and $\mathrm{h}(x), \mathrm{h}(y)$ independent for $x \neq y$.

- To compute a random hash function we have to store a table of x values and their hash values. Would take at least $O(m)$ space and $O(m)$ query time to look up $\mathrm{h}(x)$ if we hash m values. Making our whole quest for $O(1)$ query time pointless!

x	**h(x)**
x_1	45
x_2	1004
x_3	10
$\vdots$	$\vdots$
x_m	12

Efficiently Computable Hash Functions

What properties did we use of the randomly chosen hash function?

Efficiently Computable Hash Functions

What properties did we use of the randomly chosen hash function?

Pairwise Independent Hash Function. A random hash function from $\mathbf{h} : U \to [n]$ is pairwise independent if for all $i, j \in [n]$:

$$\Pr[\mathbf{h}(x) = i \cap \mathbf{h}(y) = j] = \frac{1}{n^2}.$$

$$\Pr[h(x) = i \cap h(y) = i] = \frac{1}{n^2}$$

$$\Pr[h(x) = i \cap h(y) = j \cap h(z) = k] \neq \frac{1}{n^3}$$

like for a fully random function

$$\boxed{h(x) = a \cdot x + b \bmod p}$$

Efficiently Computable Hash Functions

What properties did we use of the randomly chosen hash function?

Pairwise Independent Hash Function. A random hash function from $\mathbf{h} : U \to [n]$ is pairwise independent if for all $i, j \in [n]$:

$$\Pr[\mathbf{h}(x) = i \cap \mathbf{h}(y) = j] = \frac{1}{n^2}.$$

Exercise 1: Check the two-level hashing proof to confirm that this property is all that was needed.

Efficiently Computable Hash Functions

What properties did we use of the randomly chosen hash function?

Pairwise Independent Hash Function. A random hash function from $\mathbf{h} : U \to [n]$ is pairwise independent if for all $i, j \in [n]$:

$$\Pr[\mathbf{h}(x) = i \cap \mathbf{h}(y) = j] = \frac{1}{n^2}.$$

Exercise 1: Check the two-level hashing proof to confirm that this property is all that was needed.

When $\mathbf{h}(x)$ and $\mathbf{h}(y)$ are chosen independently at random from $[n]$, $\Pr[\mathbf{h}(x) = i \cap \mathbf{h}(y) = j] = \frac{1}{n^2}$ (so a fully random hash function is pairwise independent).

Efficiently Computable Hash Functions

What properties did we use of the randomly chosen hash function?

Pairwise Independent Hash Function. A random hash function from $\mathbf{h} : U \to [n]$ is pairwise independent if for all $i, j \in [n]$:

$$\Pr[\mathbf{h}(x) = i \cap \mathbf{h}(y) = j] = \frac{1}{n^2}.$$

Exercise 1: Check the two-level hashing proof to confirm that this property is all that was needed.

When $\mathbf{h}(x)$ and $\mathbf{h}(y)$ are chosen independently at random from $[n]$, $\Pr[\mathbf{h}(x) = i \cap \mathbf{h}(y) = j] = \frac{1}{n^2}$ (so a fully random hash function is pairwise independent).

Efficient Implementation: Let p be a prime with $p \geq |U|$. Choose random $\mathbf{a}, \mathbf{b} \in [p]$ with $\mathbf{a} \neq 0$. Represent x as an integer and let

$$\mathbf{h}(x) = (\mathbf{a}x + \mathbf{b} \mod p) \mod n.$$

Universal Hashing

Another common requirement for a hash function:

2-Universal Hash Function (low collision probability). A random hash function from $\mathbf{h} : U \to [n]$ is two universal if:

$$\Pr[\mathbf{h}(x) = \mathbf{h}(y)] \leq \frac{1}{n}.$$

fully random: $\Pr[h(x) = h(y)] = \frac{1}{n}$

Another common requirement for a hash function:

2-Universal Hash Function (low collision probability). A random hash function from $\mathbf{h} : U \to [n]$ is two universal if:

$$\Pr[\mathbf{h}(x) = \mathbf{h}(y)] \leq \frac{1}{n}.$$

Think-Pair-Shair: Which is a more stringent requirement? 2-universal or pairwise independent?

Universal Hashing

Another common requirement for a hash function:

2-Universal Hash Function (low collision probability). A random hash function from $\mathbf{h}: U \to [n]$ is two universal if:

$$\Pr[\mathbf{h}(x) = \mathbf{h}(y)] \leq \frac{1}{n}.$$

Think-Pair-Shair: Which is a more stringent requirement? 2-universal or pairwise independent?

Pairwise Independent Hash Function. A random hash function from $\mathbf{h}: U \to [n]$ is pairwise independent if for all $i, j \in [n]$:

$$\Pr[\mathbf{h}(x) = i \cap \mathbf{h}(y) = j] = \frac{1}{n^2}.$$

Universal Hashing

Another common requirement for a hash function:

2-Universal Hash Function (low collision probability). A random hash function from $\mathbf{h} : U \to [n]$ is two universal if:

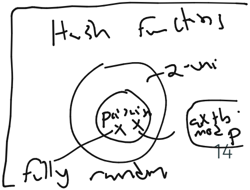
$$\Pr[\mathbf{h}(x) = \mathbf{h}(y)] \leq \frac{1}{n}.$$

Think-Pair-Shair: Which is a more stringent requirement? 2-universal or pairwise independent?

$$\Pr[\mathbf{h}(x) = \mathbf{h}(y)] = \sum_{i=1}^{n} \Pr[\mathbf{h}(x) = i \cap \mathbf{h}(y) = i] = n \cdot \frac{1}{n^2} = \frac{1}{n}.$$

$\frac{1}{n^2}$

Pairwise Ind. $\Rightarrow$ 2-universal



Universal Hashing

Another common requirement for a hash function:

2-Universal Hash Function (low collision probability). A random hash function from $\mathbf{h} : U \to [n]$ is two universal if:

$$\Pr[\mathbf{h}(x) = \mathbf{h}(y)] \leq \frac{1}{n}.$$

Think-Pair-Shair: Which is a more stringent requirement? 2-universal or pairwise independent?

$$\Pr[\mathbf{h}(x) = \mathbf{h}(y)] = \sum_{i=1}^{n} \Pr[\mathbf{h}(x) = i \cap \mathbf{h}(y) = i] = n \cdot \frac{1}{n^2} = \frac{1}{n}.$$

Remember: A fully random hash function is both 2-universal and pairwise independent. But it is not efficiently implementable.

Universal Hashing

Another common requirement for a hash function:

2-Universal Hash Function (low collision probability). A random hash function from $\mathbf{h} : U \to [n]$ is two universal if:

$$\Pr[\mathbf{h}(x) = \mathbf{h}(y)] \leq \frac{1}{n}.$$

Think-Pair-Shair: Which is a more stringent requirement? 2-universal or pairwise independent?

$$\Pr[\mathbf{h}(x) = \mathbf{h}(y)] = \sum_{i=1}^{n} \Pr[\mathbf{h}(x) = i \cap \mathbf{h}(y) = i] = n \cdot \frac{1}{n^2} = \frac{1}{n}.$$

Remember: A fully random hash function is both 2-universal and pairwise independent. But it is not efficiently implementable.

Exercise 2: Rework the two-level hashing proof to show that 2-universality is in fact all that is needed.

Questions on Hash Tables?

Next Step

$Pr(h(x)=h(y)) = n \cdot \frac{1}{n^2} = \frac{1}{n}$

pairwise

2-uni

1. We'll consider an application where our toolkit of linearity of expectation + Markov's inequality doesn't give much.
2. Then we'll show how a simple twist on Markov's (Chebyshev's inequality) can give a much stronger result.